Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*

Industrijski robot EPSON PS3 in določitev ponovljivosti

Navodila za delo pri laboratorijskih vajah iz Osnov robotike

Roman Kamnik

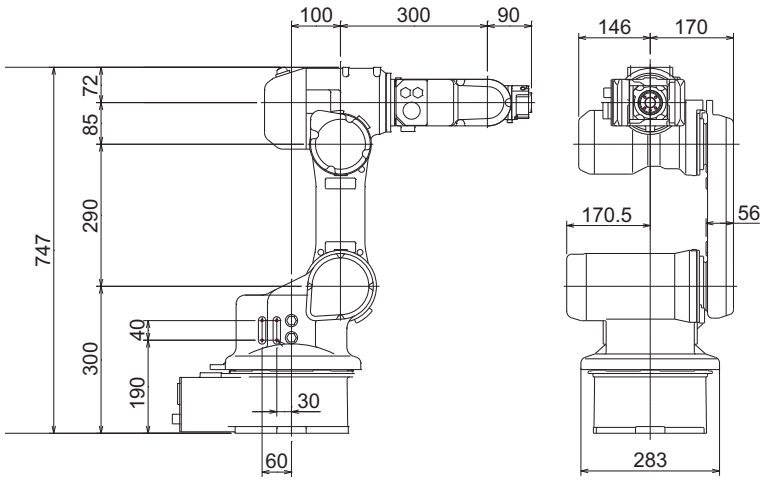
Ljubljana, 2009.

1 Robot Epson PS3

Robot Epson PS3 je robot antropomorfne konfiguracije s šestimi prostostnimi stopnjami gibanja. Robot spada v razred industrijskih robotov manjših dimenzij in je namenjen izvajanju operacij montaže ter manipuliranja z objekti. Njegova nosilnost znaša 2 kg oz. 3 kg ob določenih omejitvah izvajanja gibanja. Njegove lastnosti so:

Tip		EPSON PS3 AS00
Horizontalni Doseg		791 mm
Maksimalna hitrost	Os 1	210 $^0/s$
	Os 2	180 $^0/s$
	Os 3	225 $^0/s$
	Os 4	375 $^0/s$
	Os 5	375 $^0/s$
	Os 6	500 $^0/s$
Ponovljivost		$\pm 0.030mm$
Nosilnost nominalna/maksimalna		$2/3\ kg$
Maksimalni vztrajnostni moment	Os 4	$0.3\ kg\ m^2$
	Os 5	$0.3\ kg\ m^2$
	Os 6	$0.1\ kg\ m^2$
Maksimalni navor	Os 4	$7.25\ Nm$
	Os 5	$7.25\ Nm$
	Os 6	$5.21\ Nm$
Delovni prostor	Os 1	$+170/-170\ ^0$
	Os 2	$+150/-45\ ^0$
	Os 3	$+190/-72\ ^0$
	Os 4	$+190/-190\ ^0$
	Os 5	$+125/-125\ ^0$
	Os 6	$+360/-360\ ^0$
Teža		$45\ kg$
I/O linije		16 električnih, 2 pnevmatski
Zavore		vseh šest osi
Delovni pogoji	temperatura	$0-45^o/C$
	vlažnost	$20-80$ % RH brez kondensa
	vibracije	manj kot $4.5\ m/s^2$ (0.5G)

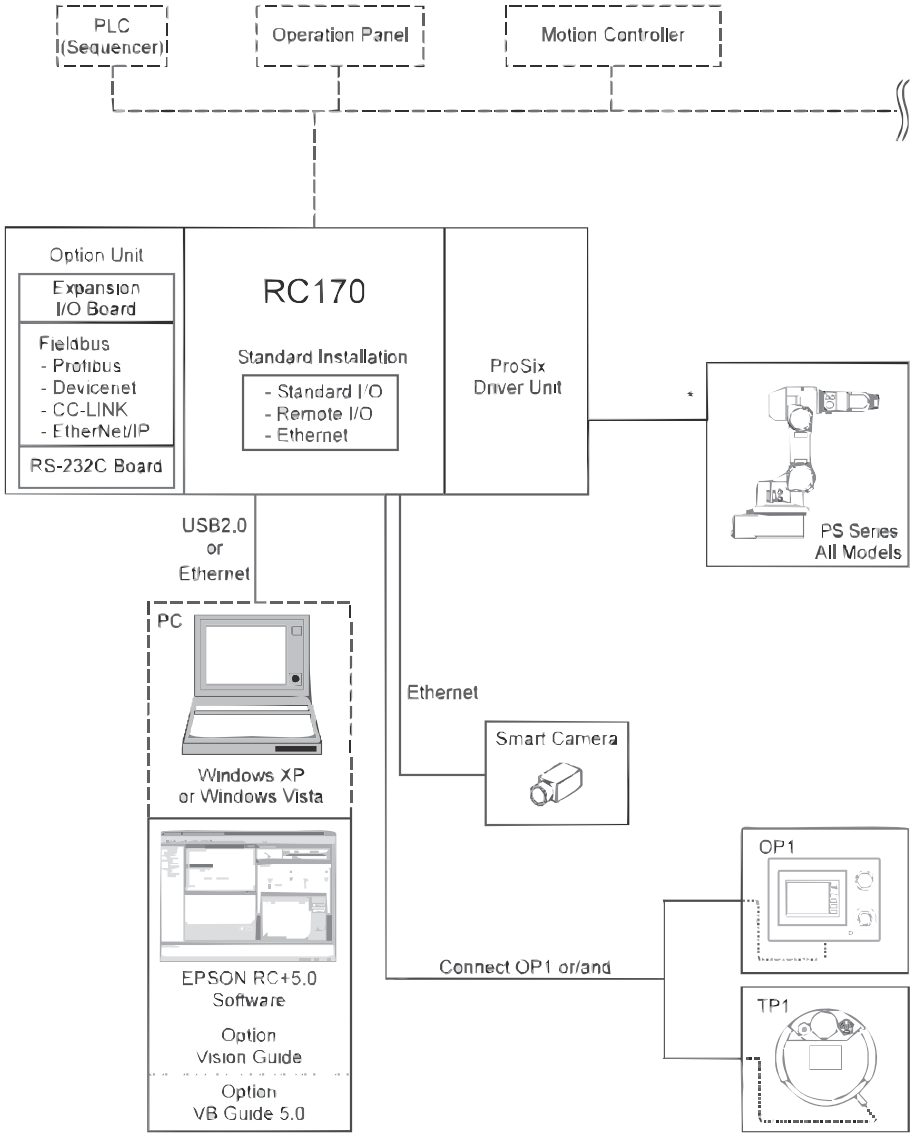
Slika 1: Fotografija robota Epson PS3 AS00



Slika 2: Dimenzije robota Epson PS3 AS00

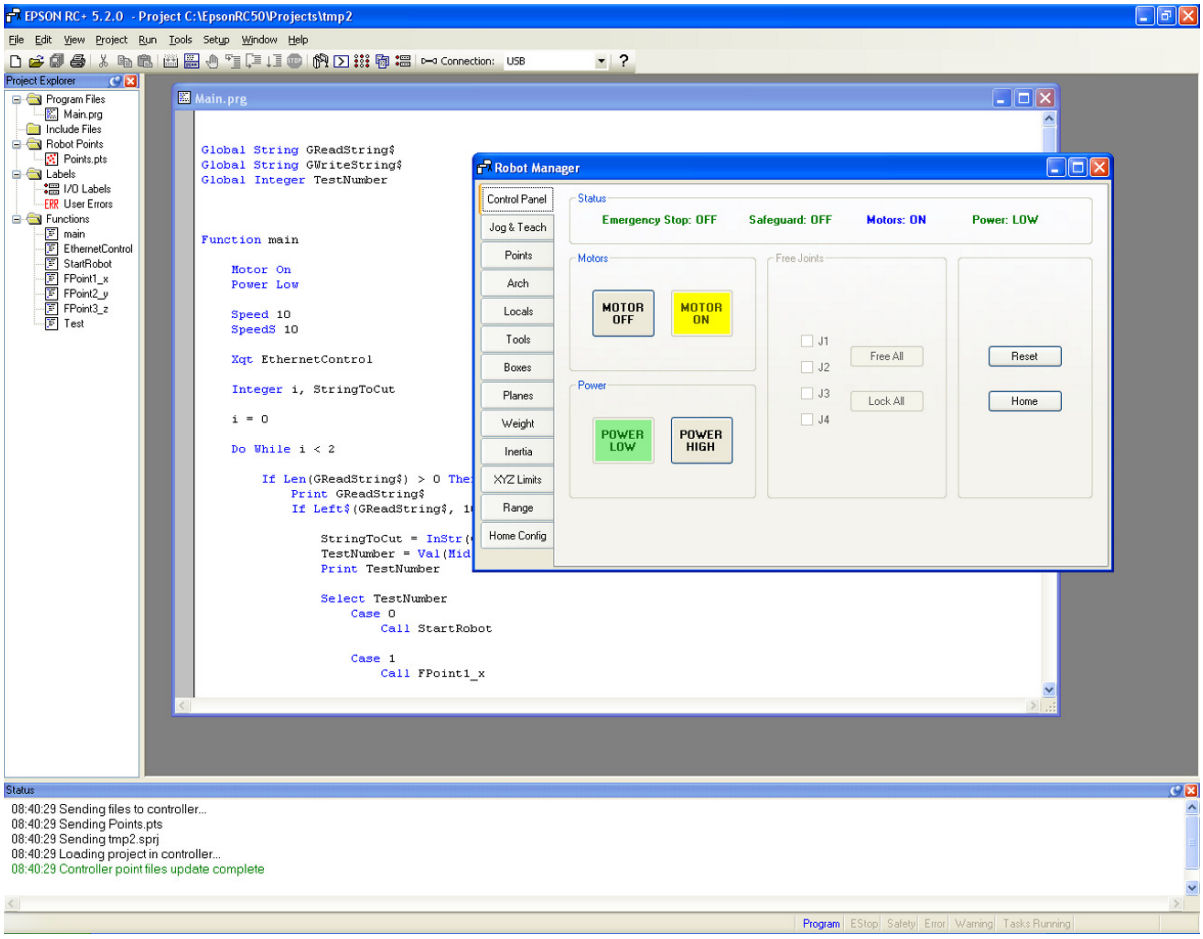
2 Robotski krmilnik Epson RC170 in programsko okolje EPSON RC+ 5.0

Krmilnik za vodenje robota EPSON RC170 je zgrajen na osnovi industrijskega osebnega računalnika, ki za vodenje posameznih osi uporablja PCI regulatorske karte. Kot uporabniški vmesnik za delo z robom služi programski paket EPSON RC+ 5.0, ki teče v operacijskem sistemu Windows XP. Programski paket omogoča ročno vodednje robota, učenje točk, razvoj programske opreme ter uporabo integriranega sistema umetnega vida. Delo s programskim paketom je enostavno in uporabniško prijazno. Na sliki 3 je prikazana principielna shema vodenja robota EPSON PS3. Sistem vodenja tvorijo krmilnik RC170, nanj priključen osebni računalnik s programskim paketom EPSON RC+ 5.0 ter opcijsko operaterski panel, ročna učilna naprava ali sistem umetnega vida.



Slika 3: Principielna shema sistema vodenja robota EPSON RS3

Na sliki 4 je predstavljeno Delovno okolje programskega paketa EPSON RC 5.0+. Okolje vsebuje brskalnik, ki vključuje vse programske strukture razvojnega projekta, delovno okno s tekstovnim urejevalnikom programa in uporabniški vmesnik *Robot Manager* za ročno vodenje in delo z robotom.



Slika 4: Delovno okolje programskega paketa EPSON RC 5.0+

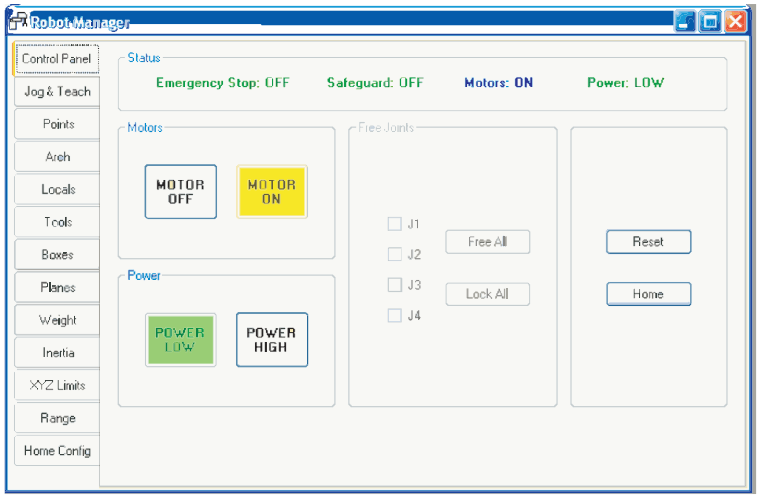
Razvoj preprostega programa

Kreiranje projekta: V meniju *Project>New project* ustvari nov projekt in ga poimenuj. Tako je ustvarjen nov projekt in program z imenom *Program.prg*.

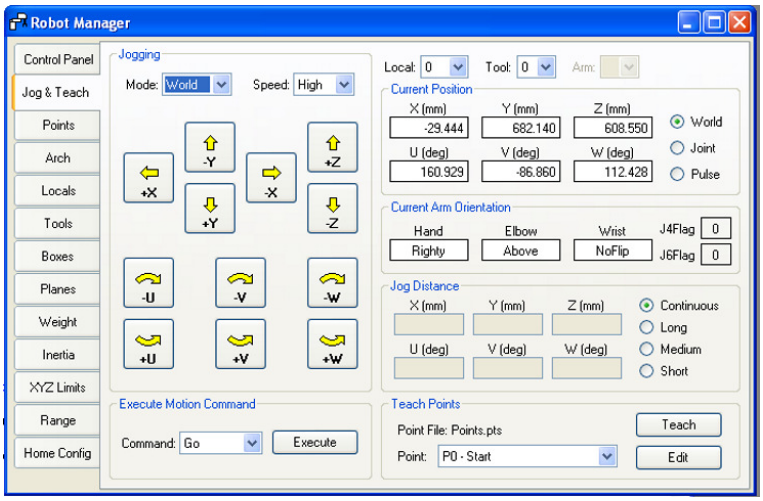
Urejanje programa: V programsko okno med stavka *Function main* in *Fend* vrivaj programske vrstice, npr. *Print "To je moj prvi program."*

Zagon programa: Z izbiro menija *Run>* opcija start (tipka F5) zaženemo prevajanje in linkanje programa, ter prenos izvršljivih datotek na krmilnik. V primeru sintaktičnih napak v programu sledi obvestilo o napaki, v primeru da je prevajanje in nalaganje uspešno, se odpre okno *Run*. Za zagon izberemo opcijo *Start*.

Učenje točk: Zagotovite neovirano gibanje robota. Izberite gumb *Robot Manager*, ki odpre okno nadzorne ploščo *Control Panel*:

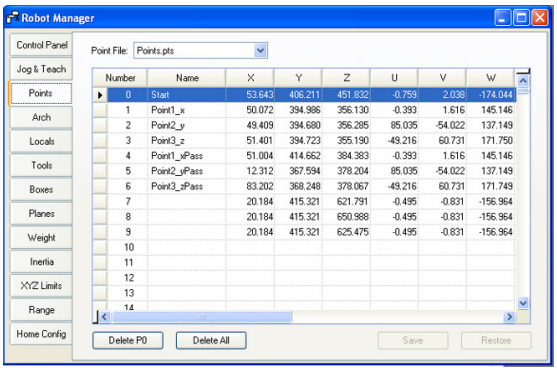


Vključite motorje z izbiro gumba *Motor On* in potrdite izbiro. Izberite *Jog & Teach* zavihek v oknu *Robot Manager*, ki odpre okno za ročno vodenje robota ter učenje točk. Učenje je možno v zunanjih koordinatah (*World*), v koordinatah sklepov (*Joint*) in koordinatah orodja (*Tool*). Z izbiro tipke (*Teach*) se trenutni položaj robota shrani pod izbranim imenom točke.



Shranjevanje in urejanje točk

Shranjene lege robota je možno urejati v urejevalniku točk (zavihek *Points* - glej sliko 5). Spreminjati je mogoče imena točk, kakor tudi posamezne koordinate. Prav tako je v urejevalniku točk možno na novo definirati točko in njene koordinate, ki so lahko izražene glede na poljuben definiran koordinatni sistem.



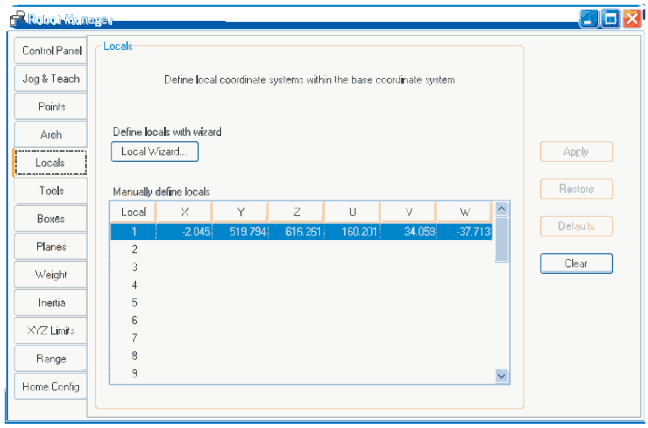
Slika 5: Urejevalnik točk

Vključevanje programskih ukazov v program: V programskem oknu dodajate programske ukaze za delovanje robotskega sistema. Spodnji primer prikazuje preprost program s katerim izvršimo izpis na ekran, nastavimo hitrost gibanja na 20 % (ukaz *Speed*), pospešek in pojemek na 20 % (ukaz *Accel*) ter programsko pošljemo robot v lego P1, lego P2 ter lego P0.

```
Function main
    Print "To je moj prvi program."
    Power High
    Speed 20
    Accel 20, 20
    Go P1
    Go P2
    Go P0
Fend
```

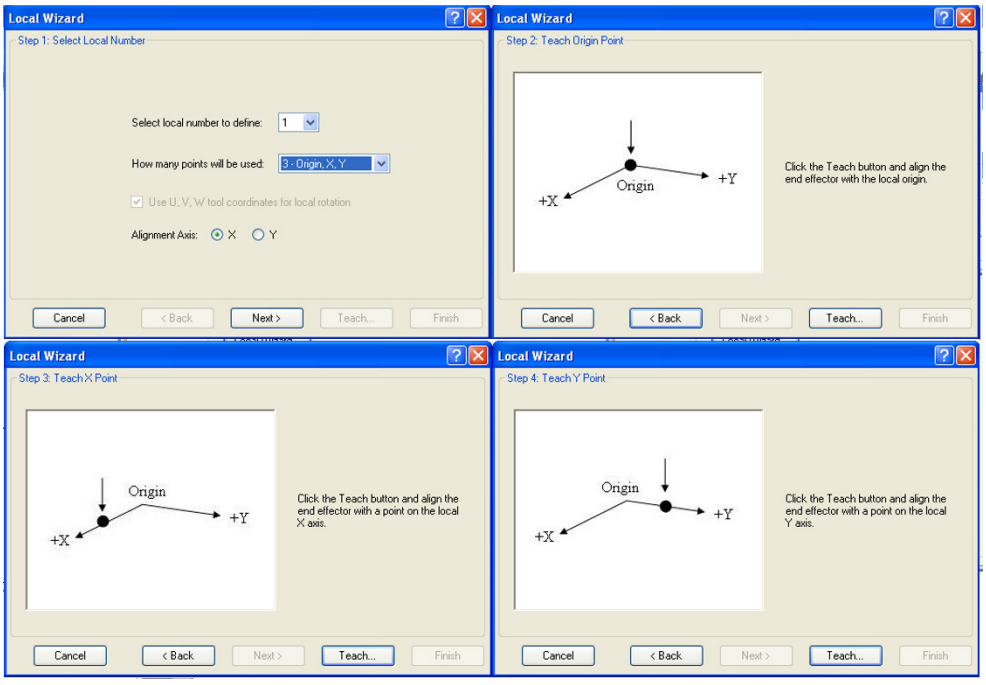
Definiranje uporabniškega koordinatnega sistema

Poljuben uporabniški koordinatni sistem v delovnem prostoru robota lahko definiramo pod zavihkom *Locals*. Z izbiro se odpre urejevalnik koordinatnih sistemov, ki dopušča urejanje, spremembo imena ali definicije koordinatnega sistema:



Slika 6: Urejevalnik koordinatnih sistemov

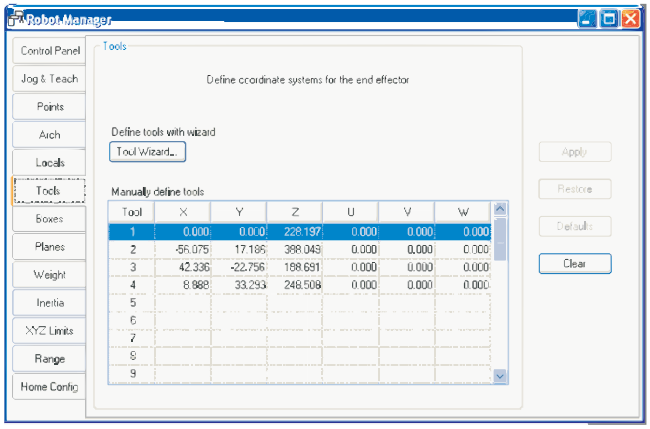
Nov koordinatni sistmem definiramo z izbiro gumba *Local Wizard*, ki je procedura, ki uporabnika vodi po korakih. V prvem oknu izbiramo ime novega koordinatnega sistem ter način kako je ta določen. Lego koordinatnega sistema določimo s tremi točkami, ki definirajo izhodišče, ter smer osi x in y koordinatnega sistema. Lego točk določimo z ročnim vodenjem vrha robota. Na spodnji sliki 7 so prikazana okna posameznih korakov za definicijo novega koordinatnega sistema. V oknih je nazorno prikazana lega točk, ki v posameznem koraku definirajo koordinatni sistem. Ko je uporabniški koordinatni sistem definiran, je možno glede nanj voditi robot (izbira opcije *Local*) ali izraziti koordinate posameznih točk.


Local Wizard
Step 1: Select Local Number
Select local number to define: 1
How many points will be used: 3 - Origin, X, Y
Use U, V, W tool coordinates for local rotation
Alignment Axis: X Y
Cancel < Back Next > Teach... Finish
Local Wizard
Step 2: Teach Origin Point
Origin
+X
+Y
Click the Teach button and align the end effector with the local origin.
Cancel < Back Next > Teach... Finish
Local Wizard
Step 3: Teach X Point
Origin
+X
+Y
Click the Teach button and align the end effector with a point on the local X axis.
Cancel < Back Next > Teach... Finish
Local Wizard
Step 4: Teach Y Point
Origin
+X
+Y
Click the Teach button and align the end effector with a point on the local Y axis.
Cancel < Back Next > Teach... Finish

Slika 7: Koraki za definiranje uporabniškega koordinatnega sistema

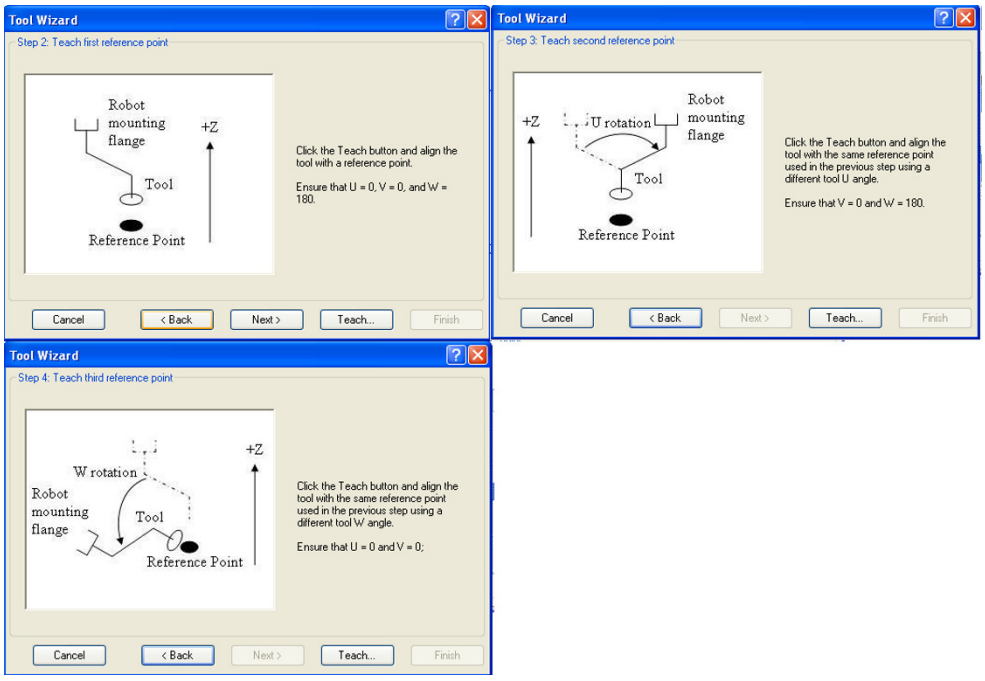
Definiranje koordinatnega sistema orodja

Kadar je na vrh robota nameščeno orodje je potrebno določiti pozicijo in orientacijo novega vrha robota, ki ga običajno postavimo v vrh orodja. Postopek, ki se imenuje kalibracija orodja, določi homogeno transformacijo med originalnim vrhom robota ter vrhom orodja. Nov koordinatni sistem orodja definiramo z izbiro menija *Tools*. Z izbiro se odpre urejevalnik koordinatnih sistemov orodij, ki dopušča urejanje, spremembo imena ali definicije koordinatnih sistemov orodja:



Slika 8: Urejevalnik koordinatnih sistemov orodja

Nov koordinatni sistem orodja določimo z izbiro gumba *Tool Wizard*, ki je procedura, ki uporabnika vodi po korakih. Nov koordinatni sistem določimo na način, da vrh orodja, ki predstavlja izhodišče novega koordinatnega sistema, ročno vodimo v isto točko s tremi različnimi orientacijami. Zahtevane orientacije so podane v oknih posameznih korakov:


Tool Wizard
Step 2: Teach first reference point
Robot mounting flange
+Z
Tool
Reference Point
Click the Teach button and align the tool with a reference point.
Ensure that U = 0, V = 0, and W = 180.
Cancel
< Back
Next >
Teach...
Finish
Tool Wizard
Step 3: Teach second reference point
+Z
U rotation
Robot mounting flange
Tool
Reference Point
Click the Teach button and align the tool with the same reference point used in the previous step using a different tool U angle.
Ensure that V = 0 and W = 180.
Cancel
< Back
Next >
Teach...
Finish
Tool Wizard
Step 4: Teach third reference point
+Z
W rotation
Robot mounting flange
Tool
Reference Point
Click the Teach button and align the tool with the same reference point used in the previous step using a different tool W angle.
Ensure that U = 0 and V = 0;
Cancel
< Back
Next >
Teach...
Finish

Slika 9: Koraki za definiranje koordinatnega sistema orodja

3 Natančnost doseganja lege industrijskega robota

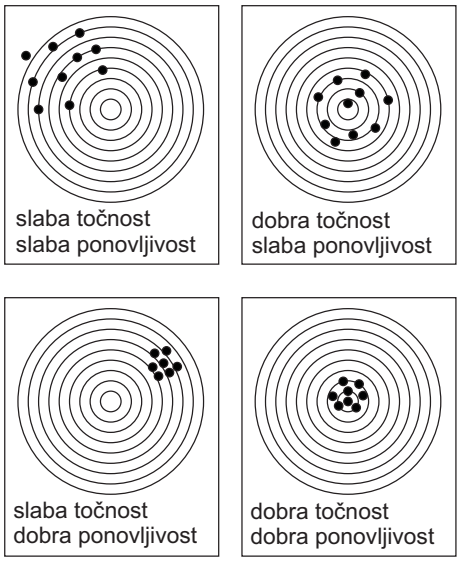
Parametri, ki določajo sposobnost pozicioniranja vrha robota v prostoru so ločljivost, točnost in ponovljivost. Na parametre vplivajo točnost kinematičnega modela (dolžine segmentov, položaj osi), mehanske lastnosti mehanizma in elektromehanske lastnosti pogonskega sistema (motorji, krmilnik).

Ločljivost je definirana kot najmanjši inkrementalni premik, ki ga robot lahko opravi. Določena je s kvaliteto regulacijskega sistema, ki vodi motorje. Predvsem je pri ločljivosti pomembna ločljivost senzorjev pozicije. Na ločljivost pa vpliva tudi mehanska zgradba robota, t.j. podajnost segmentov in mrtvi hod v sklepih.

Točnost označuje robotovo zmožnost, da pozicionira vrh v določeno lego znotraj delovnega prostora. Točnost je odvisna od mehanskih lastnosti mehanizma, algoritmov vodenja in ločljivosti. Točnost je možno izboljšati z natančnejšo izdelavo segmentov, s povečanjem togosti mehanizma (vendar povečanje mase lahko upočasni robot), z uporabo kvalitetnejših senzorjev in regulacijskega sistema. Pomemben dejavnik pri zagotavljanju absolutne točnost doseganja lege robota je tudi razlika med nično pozicijo sklepov, zapisano v krmilniku ter dejansko nično pozicijo sklepov. Točnost se spreminja glede na pozicijo vrha manipulatorja v delovnem prostoru, saj so npr. obremenitve sklepov večje pri bolj iztegnjenem robotu.

Ponovljivost je zmožnost robota, da dosega določeno lego vedno znova. Je statistični pojem, povezan z natančnostjo in opisuje kakšne so napake pri ponavljanju istih gibov. Če robot programsko vodimo iz iste lege v merjeno lego, vedno pri enakih vplivih okolja, z določenim številom ponovitev, lahko opazimo, da rezultati kažejo odstopanje od pričakovane vedno iste vrednosti.

Grafično lahko parametra ponovljivosti in točnosti ponazorimo s sliko 11.



Slika 10: Točnost in ponovljivost

Absolutna točnost doseganja lege je pomembna pri zamenjavah robotov in indirektnem programiranju preciznih aplikacij. Pri indirektnem programiranju je robotski program razvit v simulacijskem okolju in samo prenešen v robotski krmilnik. Natančnost izvajanja programa je odvisna od ujemanja modela robotske celice in realne celice ter od točnosti doseganja leg robota. Ponovljivost pa je pomembna pri direktnem programiranju robota, kjer robot vodimo od točke do točke in lege shranjujemo v krmilnik. Robot nato med delovanjem izvaja gibanje vzdolž ročno določene trajektorije. Večina proizvajalcev robotov v tehničnih karakteristikah svojih robotov raje podaja podatek o ponovljivosti kot o natančnosti.

4 Merjenje ponovljivosti industrijskega robota

Ponovljivost opisuje deviacijo povprečja napak pri doseganju določene lege. Ponovljivost ni napaka, ki bi bila izražena glede na bazni koordinatni sistem, ampak je to deviacija povprečja napak. Izraža razpršenost lege pri premiku robota v zahtevano lego n-krat iz iste smeri. Mednarodni standard ISO9283 definira pomembne karakteristike, ki določajo zmogljivost robotov, opisuje na kakšen način naj bodo določene ter predlaga, kako jih izmeriti. Ena izmed karateristik je ponovljivost pozicije vrha robota, ki je po standardu ISO9283 definirana z enačbo:

$$r = \bar{l} \; + \; 3s \tag{1}$$

kjer je

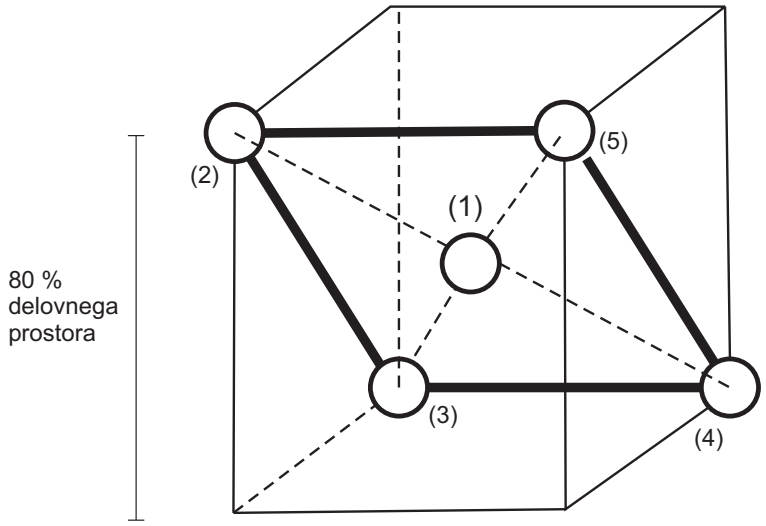
$$\bar{l} \;\; = \;\; \frac{1}{n} \sum_{i=1}^{n} l_i \tag{2}$$

$$l_i \;\; = \;\; \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2} \tag{3}$$

$$s \;\; = \;\; \sqrt{\frac{\sum_{i=1}^{n} (l_i - \bar{l})^2}{n-1}} \tag{4}$$

V enačbi (3) nastopajo členi x_i, y_i, z_i, ki predstavljajo posamezne odčitke koordinat pozicije vrha robota v določeni točki, $\overline{x}, \overline{y}, \overline{z}$ pa so povprečne vrednosti vseh odčitkov.

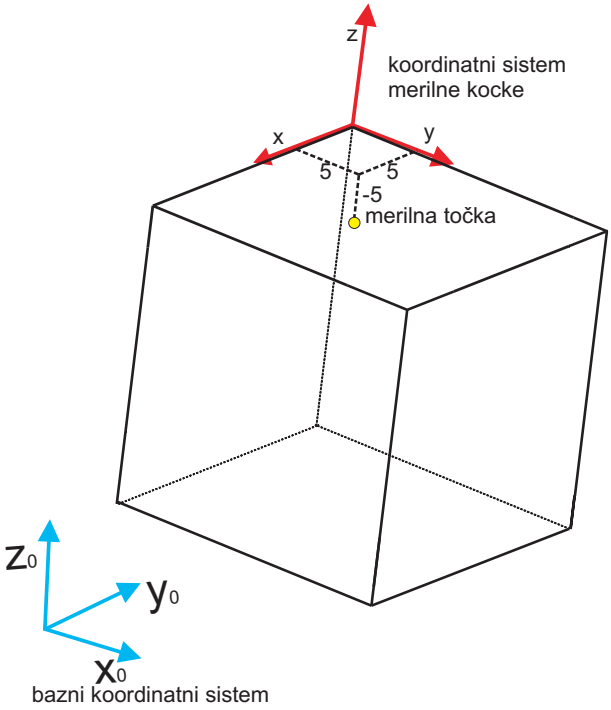
Standard ISO9283 predpisuje merilne pogoje, ki vključujejo zahteve glede električnih motenj, temperature in ostalih vplivov okolja. Standard predpisuje, da mora robot biti v stabiliziranih pogojih delovanja, meritve pa morajo biti izvedene s 100% obremenitvijo (opcijsko tudi z 10%) ter hitrostjo gibanja 100%, 50% in 10% deklarirane najvišje hitrosti. Meritve se izvajajo določenih točkah delovnega prostora, ki so definirane s t.i. ISO kocko s predpisanim številom ponovitev. ISO kocka je kocka znotraj delovnega prostora robota, kjer se pričakuje največ gibanja robota med delovanjem. Dolžina stranic kocke mora biti maksimalna glede na delovno področje robota. Na sliki 11 je prikazana varianta merilnih točk določenih z ISO kocko v delovnem prostoru robota.



Slika 11: Merilne točke v delovnem prostoru robota v katerih določamo ponovljivost

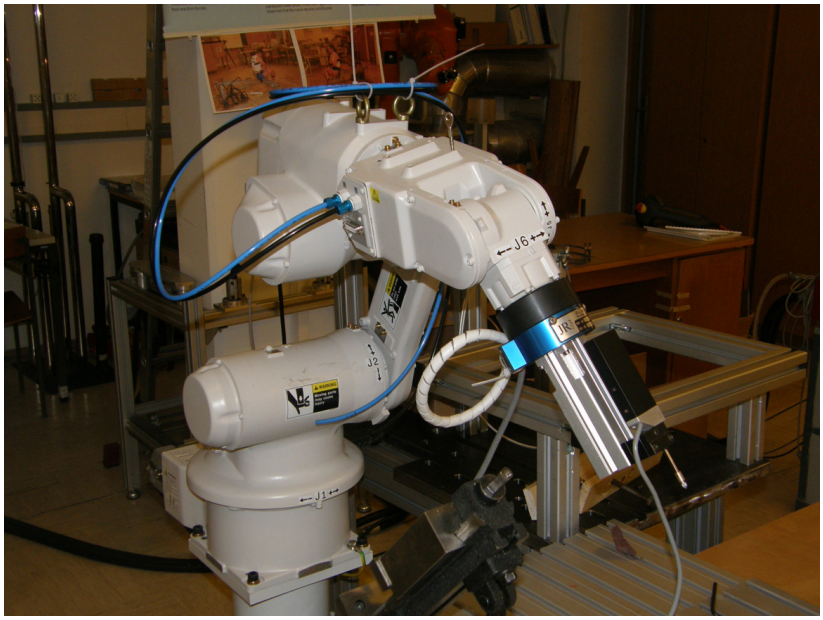
5 Navodila za izvedbo vaje

Naloga zahteva določitev ponovljivosti robota EPSON RS3 v eni točki delovnega prostora pri dveh obremenitvah. Merilna točka je določena v notranjosti kovinske kocke nameščene v delovnem prostoru robota. Potrebno je izmeriti ponovljivost v točki, ki je $5\ mm$ oddaljena od robov kocke, kot to prikazuje slika 12.



Slika 12: Merilna točka v notranjosti merilne kocke in lokalni koordinatni sistem pripet na kocko

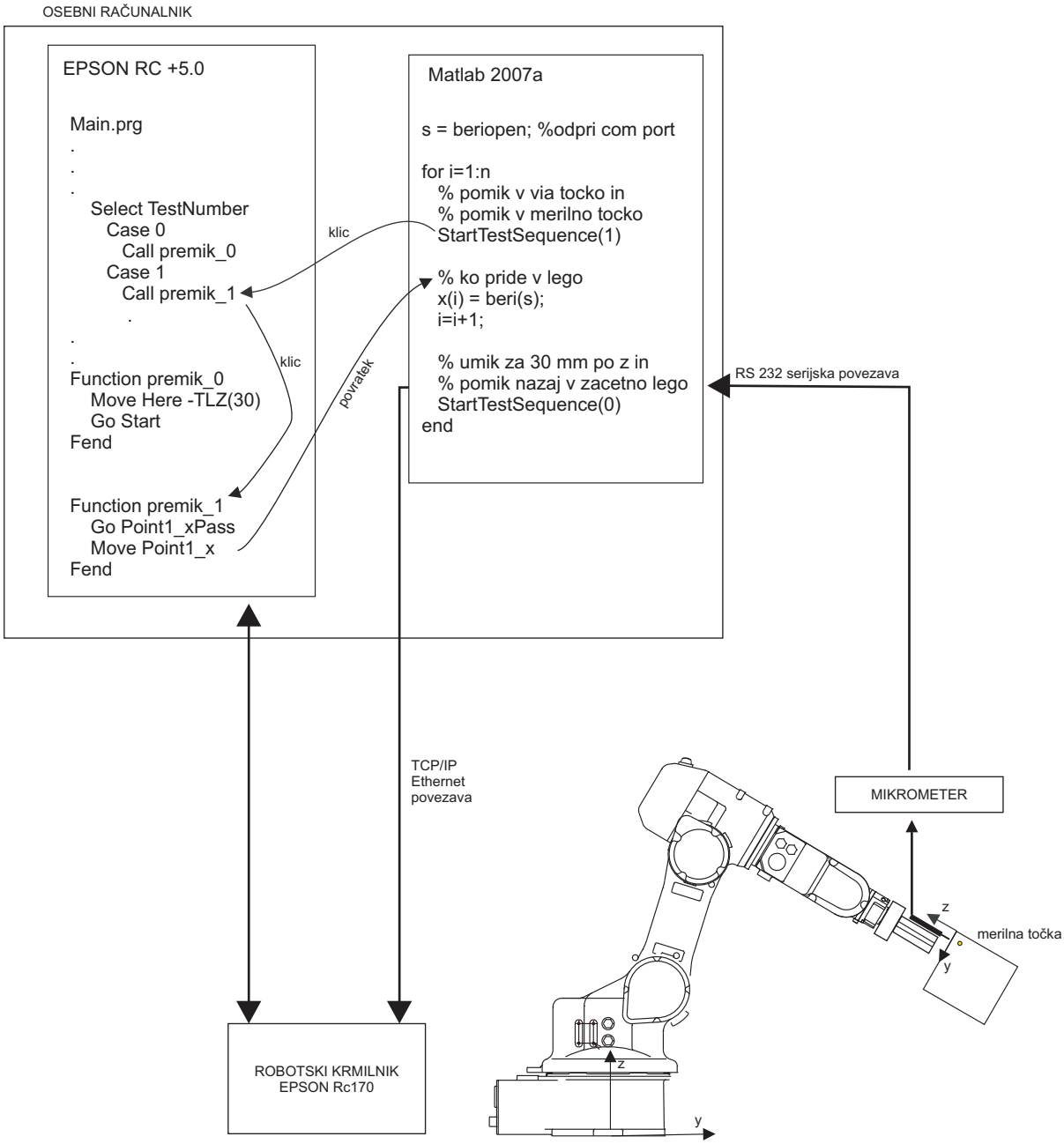
Pri izvedbi preciznih meritev ponovljivosti pozicioniranja robota uporabite elektronski mikrometer, ki meri premik merilne konice vzdolž ene dimenzije. Mikrometer je na robot nameščen kot kaže slika 13. V zapestju robota je nameščen štiriosni senzor sil in momenta, ki v primeru prevelikih obremenitev mikrometra izključi motorje robota.



Slika 13: Fotografija merilnega sistema za merjenje ponovljivosti

Ker uporabljeni merilni sistem meri premik le vzdolž ene dimenzije, je za določitev ponovljivosti potrebno opraviti tri serije meritev, ločeno za vsako smer posebej. V vsaki seriji meritev robot opravi predpi-

sano število ponovitev pozicioniranja v točko iz določene smeri (x, y, ali z). Ko se pri posamezni ponovitvi vrh robota ustali v ciljni legi, je potrebno odčitati vrednost premika merilne konice mikrometra.



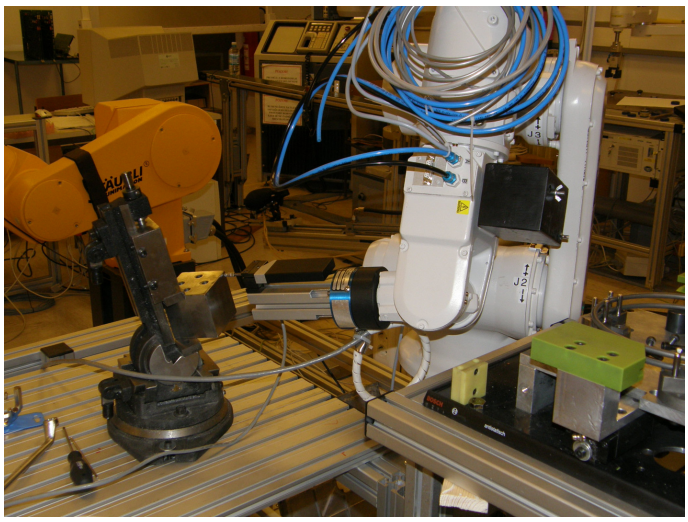
Slika 14: Blokovna shema merilnega sistema za določitev ponovljivosti robota

Na sliki 14 je prikazana blokovna shema merilnega sistema za določitev ponovljivosti. Robot je voden s pomočjo robotskega krmilnika EPSON RC170 in programskega paketa EPSON RC+5.0, ki teče na osebnem računalniku. Izmerjene podatke z mikrometra zajemamo v okolju Matlab (verzija 2007a), ki teče na istem osebnem računalniku. Okolje Matlab z nadzorno enoto mikrometra komunicira preko serijske povezave RS232. Klic funkcije *beri* vrne trenutno vrednost premika mikrometra. Komunikacija med okoljem Matlab in robotskim krmilnikom pa je izvedena preko TCP/IP povezave s pomočjo uporabe funkcije *StartTestSequence*, ki robotskemu krmilniku posreduje celoštevilski parameter. Glede na vrednost parametra robotski program izvede vejitev in skok na ustrezno funcijo, ki vsebuje zaporedje ukazov za gibanje. Na sliki 14 sta v dveh blokih osebnega računalnika prikazana primera programske kode v okolju Matlab in RC+5.0 za izvedbo serije n merilnih ponovitev izvedenih iz npr. smeri x.

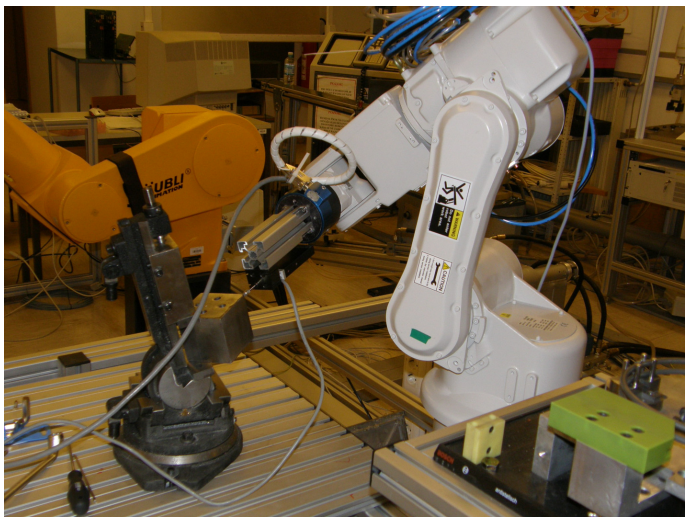
Priporočila za določitev točk

Za izvedbo premikov v merilno točko iz treh različnih smeri je potrebno:

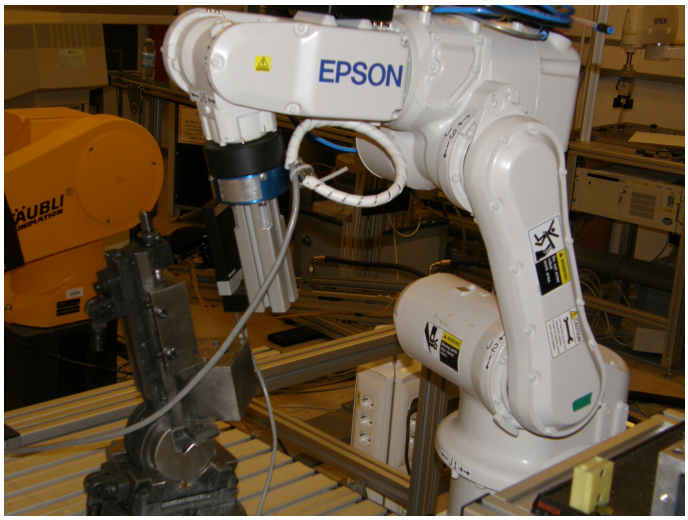
- definirati koordinatni sistem orodja *Tool*, ki je pripet na vrhu konice mikrometra. Glej razdelek *Definiranje koordinatnega sistema orodja*.
- definirati lokalni koordinatni sistem *Local* glede na robove kocke, kot to prikazuje slika 12. Glej razdelek *Definiranje uporabniškega koordinatnega sistema*.
- glede na koordinatni sistem kocke določiti koordinate točk približevanja in končne lege za tri različne orientacije. Spodnje tri slike prikazujejo tri primere približevanja merilni točki iz smeri x, y in z glede na koordinatni sistem kocke. Za posamezne točke je navedena pozicija in orientacije prijemala ter parametri robotske konfiguracije.



Slika 15: Izvajanje meritve ponovljivosti vzdolž smeri x:
merilna točka: *[5, 5, -5, 0, -90, 180]*
točka približevanja: *[-20,5,-5,0,-90,180, Righty, Above, NoFlip, J6 flag=1]*



Slika 16: Izvajanje meritve ponovljivosti vzdolž smeri y:
merilna točka: *[5, 5, -5, 90, 90, 0]*
točka približevanja: *[5,-20,-5,90,90,0, Righty, Above, NoFlip, J6 flag=0]*



Slika 17: Izvajanje meritve ponovljivosti vzdolž smeri z:
merilna točka: *[5, 5, -5, 0, 0, 180]*
točka približevanja: *[5,5,20,0,0,180, Righty, Above, NoFlip, J6 flag=1]*